

Toepassingen en Algoritmes van Geavanceerde Programmeertalen

Taak “Functioneel Programmeren in Erlang” – 2017-‘18

Algemeen

De benaming van het opleidingsonderdeel zegt het al toepassingEN en programeertalEN. Er zijn dus verschillende componenten in de evaluatie met verschillende evaluatievormen

Evaluatie voornamelijk via een taak, met eventueel toelichting op het examen

1. Functioneel programmeren in Erlang
2. Multi-core programmeren met het Actor-model van Erlang/OTP
3. Java (met lambda's, stream() en JavaFX)

Evaluatie volledig op het examen

4. Erlang versus Haskell
5. Tijdscomplexiteit

Dit document beschrijft énkél de opdracht voor de eerste taak (FP in Erlang)

Een van de geneugtes van het studeren aan de universiteit is dat je op kot kan gaan: je kan veel meer je eigen ding doen en je eigen planning volgen, maar je moet natuurlijk ook wel zelf voor je eten zorgen: kortom je wordt (langzaam) zelfstandig.

Aan de keerzijde hiervan is er wel het feit dat het kotleven duur is. Je moet maandelijks huur betalen, meestal m.i.v. een voorschot voor de energiekosten. Die worden op het einde van het jaar wel verrekend met de reële kosten, en het kan dan zijn dat je geld terug krijgt, maar ook dat je moet bij betalen.

In deze taak moet je allerlei berekeningen doen i.v.m. de boekhouding van zo'n studentenkot. Hierbij is er in de eerste plaats een lijst met alle verrichtingen voor een studentenhuus met daarin verschillende koten. De volgende verrichtingen komen voor in willekeurige volgorde:

- Een betaling door een student voor een specifieke kamer. Hierbij wordt opgegeven
 - o Datum van betaling
 - o Nummer van de kamer
 - o Betaald bedrag
- Een korting die toegekend is door de kotbaas, bijvoorbeeld omdat de verwarming in een bepaalde maand defect was en het een aantal dagen duurde voordat dit hersteld was. Hierbij wordt opgegeven:
 - o Datum van de toeslag
 - o Nummer van de kamer
 - o Te betalen bedrag
 - o Reden voor de korting
- Een toeslag aangerekend door de kotbaas, bijvoorbeeld omdat de student dingen kapot had gemaakt of een boete wegens inbreuken tegen het huisreglement. Dit omvat:
 - o Datum van de toeslag
 - o Nummer van de kamer
 - o Te betalen bedrag
 - o Reden voor de korting

- De afrekening van de energiekosten, met volgende gegevens:
 - o Datum van de afrekening
 - o Nummer van de kamer
 - o Totaal bedrag te betalen voor de energiekosten.
 Let op! Dit is niet het bedrag dat de student moet bijbetalen of terugkrijgt. De voorschotten voor energie moeten hier van afgetrokken worden.

Daarnaast is er ook nog een lijst met de contracten voor de studentenkamers. Hierin staan per kamer volgende gegevens:

- Nummer van de kamer
- Naam van de eigenaar van de kamer
- Naam van de huurder
- Prijs per maand voor de huur
- Prijs per maand voor het voorschot van de kosten van energie
- Datum aanvang contract
- Het aantal huurmaanden van dit contract.

Merk dus op dat er in hetzelfde studentenhuus koten van verschillende eigenaars kunnen zijn.

Het is de bedoeling dat je de nodige data-types definieert en een aantal leuke/interessant functies implementeert. Je mag zelf kiezen welke functies je maakt, maar het is natuurlijk wel de bedoeling dat je niet alleen oersimpele functies maakt. Het concept is dat je een soort showcase maakt van hoe goed je kan functioneel programmeren in Erlang, met focus op functioneel en recursief denken en lijst-bewerkingen zonder neveneffecten. Zoals het hoort bij een mastervak is er een grote mate van vrijheid en ligt enkel het onderwerp op zich vast, maar wat je er precies mee doet, kan en moet je zelf aanvullen. De belangrijkste voorwaarde is dat je er voor zorgt dat je laat zien dat je op elegante en efficiënte code kan schrijven in Erlang volgens de regels van de kunst van het functioneel programmeren. De evaluatie zal dan ook meer gaan over het kunnen toepassen van de regels van die kunst, dan over de eigenlijke inhoud van de functies. Toch zijn er een aantal functies die redelijk vanzelfsprekend zijn en die we dan ook min of meer verwachten in jullie oplossing:

- Een functie die gegeven die het aantal kamers per eigenaar terug geeft
- Een functie die gegeven de lijst met contracten, de lijst met verrichtingen en het nummer van een kamer, terug geeft hoeveel de student nog moet bijbetalen of terug krijgen.
- Een functie die gegeven de lijst met contracten en de lijst met verrichtingen, terug geeft hoeveel inkomsten elke eigenaar heeft.

Daarbuiten is het enige wat eigenlijk vast ligt de tijdsbesteding. Deze taak staat op 22,50% van de punten van TAGP en je moet dus 22,50% van de tijd aan deze taak besteden (aan 6 studiepunten en 25 tot 30u per studiepunt is 22,50% gelijk aan 34 tot 41u).

Deadline voor deze taak: donderdag 9/11/2017 om 20u

Veel succes!
Kris Aerts, Stijn Schildermans